

VŠB – Technická univerzita Ostrava
Fakulta strojní
Katedra automatizační techniky a řízení

Řízení distribuovaných systémů pomocí PLC
Control of Distributed Systems using PLC

Student:

Bc. Jakub Novotný

Vedoucí diplomové práce:

Ing. Jaromír Škuta, Ph.D.

Zadání diplomové práce

Student: **Bc. Jakub Novotný**
Studijní program: **N2301 Strojní inženýrství**
Studijní obor: **3902T004 Automatické řízení a inženýrská informatika**
Téma: **Řízení distribuovaných systémů pomocí PLC**
Control of Distributed Systems using PLC

Zásady pro vypracování:

1. Seznamte se s problematikou distribuovaných systémů řízení a proveďte návrh distribuovaného systému řízení modelové úlohy v podmínkách laboratoře.
2. Realizujte navrženou modelovou úlohu v podmínkách laboratoří katedry, vyberte vhodné řídicí systémy a navrhnete způsob vzájemné komunikace.
3. Vytvořte potřebné algoritmy pro vybrané řídicí systémy. Pro monitorování a konfiguraci této úlohy využijte SCADA/MMI systém Control Web 6.1.
4. Zhodnoťte dosažené výsledky a navrhnete směr dalšího řešení.

Seznam doporučené odborné literatury:

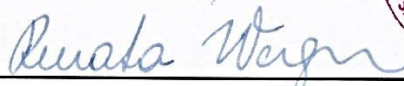
BALÁTĚ, J. *Automatické řízení*. Praha: Nakladatelství BEN, 2003, 654 s. ISBN 80-7300-020-2.
CNC Výukové kity [ONLINE]. HELAGO-CZ, s.r.o. 2002-2012. [CIT. 11.10.2012]. Dostupné z: <http://www.helago-cz.cz/catalog/cnc-vyukove-kity/>.
JANEČEK, J. 1993. *Distribuované systémy*. Praha : Vydavatelství ČVUT, 1993.
VAEPROSYS - *Automatizace - PLC AC500* - Dokumentace ke stažení [ONLINE]. VAE PROSYS S.R.O. 2010. [CIT. 11.10.2012]. Dostupné z: <http://vaeprosys.cz/list.asp?P=133&K=32&MK=9&MS=32>.
VLACH, J. *Počítačová rozhraní, přenos dat a řídicí systémy*. PRAHA, BEN-Technická literatura, 1997, ISBN 80-85940-17-4.


Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jaromír Škuta, Ph.D.**

Datum zadání: 13.12.2014
Datum odevzdání: 18.05.2015




doc. Ing. Renata Wagnerová, Ph.D.
vedoucí katedry


doc. Ing. Ivo Hlavatý, Ph.D.
děkan fakulty

Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě dne 18. 5. 2015

.....
podpis studenta

Prohlašuji, že

- jsem byl seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě diplomovou práci užít (§ 35 odst. 3).
- souhlasím s tím, že diplomová práce bude v elektronické podobě uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě: 18. 5. 2015

.....
podpis

Jméno a příjmení autora práce: Bc. Jakub Novotný

Adresa trvalého pobytu autora práce: 17. listopadu 32, Havířov – Podlesí, 736 01

ANOTACE DIPLOMOVÉ PRÁCE

NOVOTNÝ, J. *Řízení distribuovaných systémů pomocí PLC*. Ostrava, 2015. 45 s.
Diplomová práce. Strojní fakulta VŠB-TUO. Katedra automatizační techniky a řízení.
Vedoucí diplomové práce: Škuta, J.

Tato práce se zabývá řízením distribuovaného systému pomocí PLC. Jsou zde popsána použitá PLC od firmy ABB, možnosti jejich komunikace a jejich vývojové prostředí CODESYS. Pro komunikaci mezi jednotlivými PLC byl použit protokol Modbus RTU. Propojení bylo realizováno přes standard sériové komunikace RS485. Na nadřazeném PLC je program pro ovládání modelu výtahu a ovládání komunikace přes protokol Modbus RTU. Na podřízených PLC jsou ovládány úlohy realizující řízení stejnosměrného motoru a tlakovzdušné soustavy a úloha realizující vrtání plošných spojů. Pro tento systém byla vytvořena i ovládací a monitorovací aplikace v programu Control Web 6. Komunikace mezi PLC a programem Control Web 6 je realizována pomocí OPC protokolu.

ANOTATION OF DIPLOMA THESIS

NOVOTNÝ, J. *Control of Distributed Systems using PLC*. Ostrava, 2015. 45 p. Diploma
Thesi. Faculty of Mechanical Engineering VŠB-TUO. Department of Control Systems
and Instrumentation. Thesis Supervisor: Škuta, J.

This work is dealing with distributed system control. PLCs from ABB that are used are described here, as well as possibilities of their communication and their developing interface CODESYS. The Modbus protocol was used for communication between PLCs. The connection was realised via RS485 standard. In master PLC there is a program, which controls the model of an elevator. In slave PLCs there are programs for control of DC engine and compressed air system and the program which deals with drilling of printed circuit boards. There was also created an application for monitoring and control in Control Web 6. The communication between PLC and Control Web 6 was carried out via OPC protocol.

Obsah

Seznam použitých značek a symbolů.....	8
Úvod.....	10
1 Návrh distribuovaného systému.....	11
1.1 Systémový návrh.....	11
1.2 Stejnoseměrný motor.....	12
1.3 Tlakovzdušná soustava	13
1.4 Model výtahu	14
1.5 Vrtání plošných spojů	15
1.6 Použitá PLC	17
2 CODESYS.....	20
2.1 Programovací prostředí a komunikace s PLC.....	20
2.2 Programovací jazyky	21
3 Control Web 6.....	24
3.1 Vývojové prostředí	24
3.2 Grafický editor.....	25
3.3 Datové inspektory	27
3.4 Časování.....	28
4 Komunikace	29
4.1 Modbus RTU	29
4.2 OPC komunikace	31
5 Tvorba aplikací.....	34
5.1 Stejnoseměrný motor.....	34
5.1.1 Vysokorychlostní čítač	34
5.1.2 Dolní propust	35
5.1.3 Řízení.....	35
5.1.4 Ovládací a monitorovací aplikace	36
5.2 Tlakovzdušná soustava	36

5.3	Model výtahu	37
5.4	Vrtání plošných spojů	38
5.4.1	Ovládací a monitorovací aplikace	38
5.4.2	Řízení krokových motorků	40
6	Závěr	42
	Seznam použité literatury	44

Seznam použitých značek a symbolů

ABB	Firma poskytující technologie pro energetiku a automatizaci (ASEA Brown Boveri)
ADU	Aplikační úroveň (Aplication Data Unit)
CFC	Jazyk volně propojovaných bloků (Continuous Function Chart)
CNC	Počítačové číslicové řízení (Computer Numerical Control)
CS31	Komunikační protokol firmy ABB
FBD	Jazyk funkčního blokového schématu (Function Block Diagram)
FBP	Fieldbus konektor (Fieldbus Plug)
HMI	Rozhraní mezi člověkem a strojem (Human Machine Interface)
I/O	Vstup/výstup (Input/Output)
IL	Instrukční sada (Instruction List)
IP	Internetový protokol (Internet Protocol)
IRC	Inkrementální rotační snímač (Incremental Rotation enCoder)
LD	Jazyk žebříčkového diagramu (Ladder Diagram)
OPC	Komunikační protokol (OLE for Process Control)
PC	Osobní počítač (Personal Computer)
PDU	Úroveň protokolu (Protocol Data Unit)
PIC	Mikrokontrolér (Peripheral Interface Controller)
PLC	Programovatelný logický automat (Programmable Logic Controller)
PWM	Pulzně šířková modulace (Pulse Width Modulation)
RS232	Standart sériové komunikace
RS485	Standart sériové komunikace
RTS	Požadavek na poslání (Request To Send)
RTU	Vzdálená koncová jednotka (Remote Terminal Unit)
SCADA	Dispečerské řízení a sběr dat (Supervisory Control And Data Aquisition)
SFC	Sekvenční funkční diagram (Sequential Function Chart)
ST	Jazyk strukturovaného textu (Structured Text)
UDP	Druh internetového protokolu (User Datagram Protocol)

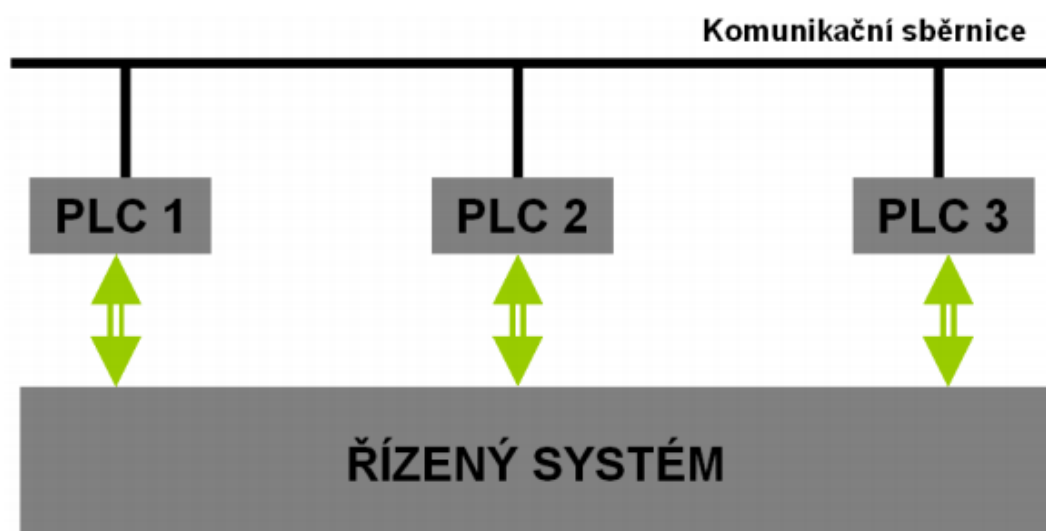
G_f Přenos filtru
 s komplexní proměnná, nezávisle proměnná u obrazu v
Laplaceově transformaci

Úvod

V dnešní době si už život bez automatizace nelze ani představit. Díky automatizaci dochází k úspoře lidských zdrojů a spousta procesů by bez prostředků automatického řízení nebyla ani možná. Spousta automatizovaných systémů je natolik komplexní, že realizace pomocí jednoho řídicího PLC by byla velmi složitá, anebo by ani nebylo možné takto projekt realizovat např. kvůli velkým vzdálenostem mezi jednotlivými řízenými podsystémy. V tu chvíli přichází na scénu systém distribuovaného řízení, kde jsou dílčí úlohy rozděleny mezi více řídicích jednotek, které mezi sebou komunikují po komunikačních sběrnících a mohou být ovládány nadřazeným řídicím systémem. Takovéto systémy jsou pak i snáze rozšiřitelné. A právě tvorbou systému distribuovaného řízení se tato práce zabývá.

1 Návrh distribuovaného systému

Distribuovaný systém je složen z několika dílčích systémů, které jsou propojeny komunikační sběrnicí a společně se podílejí na řízení. Tímto rozdělením činností mezi jednotlivé dílčí systémy je umožněno dosáhnout vyšších výkonnostních možností a výstavba složitějších systémů se tak stává jednodušší. Takovýto systém je pak možné rozšiřovat přidáním dalšího zařízení na komunikační sběrnici. [Koziorek a spol., 2011]



Obr. 1 - Struktura distribuovaného řídicího systému [Koziorek a spol., 2011]

1.1 Systémový návrh

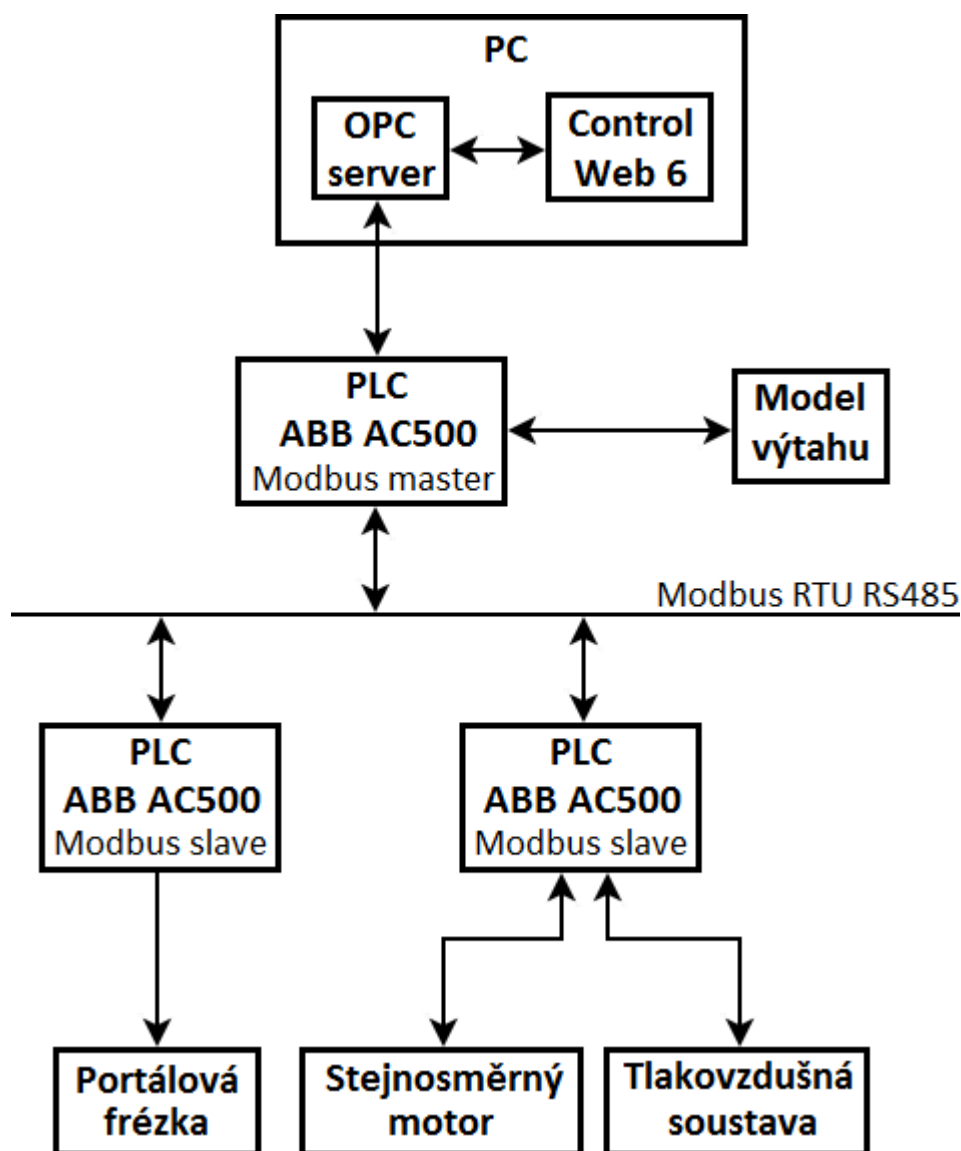
Účelem této práce je vytvoření distribuovaného systému v laboratorních podmínkách, kdy jednotlivé systémy budou řízeny pomocí PLC.

Pro tento systém byly vybrány následující laboratorní úlohy.

- Stejnosměrný motor
- Tlakovzdušná soustava
- Model výtahu
- Vrtání plošných spojů

Byla vybrána PLC firmy ABB, která mezi sebou komunikují pomocí protokolu Modbus RTU. Jedno PLC je řídicí. Tedy posílá podřízeným PLC příkazy a rovněž si stahuje data informující o řízených úlohách. Modbus RTU je master/slave protokol, takže podřízená PLC mezi sebou nemohou komunikovat přímo. Tato PLC se programují pomocí

softwaru CODESYS. Mimo jiné CODESYS obsahuje OPC server, jímž je zprostředkována výměna dat mezi nadřazeným PLC a ovládací a monitorovací aplikací vytvořenou v prostředí softwaru Control Web 6. Nadřazené PLC pak kromě komunikace ovládá model výtahu. Jedno podřízené PLC ovládá portálovou frézku, na níž probíhá vrtání plošných spojů. Druhé podřízené PLC pak najednou ovládá řízení stejnosměrného motoru a tlakovzdušné soustavy.



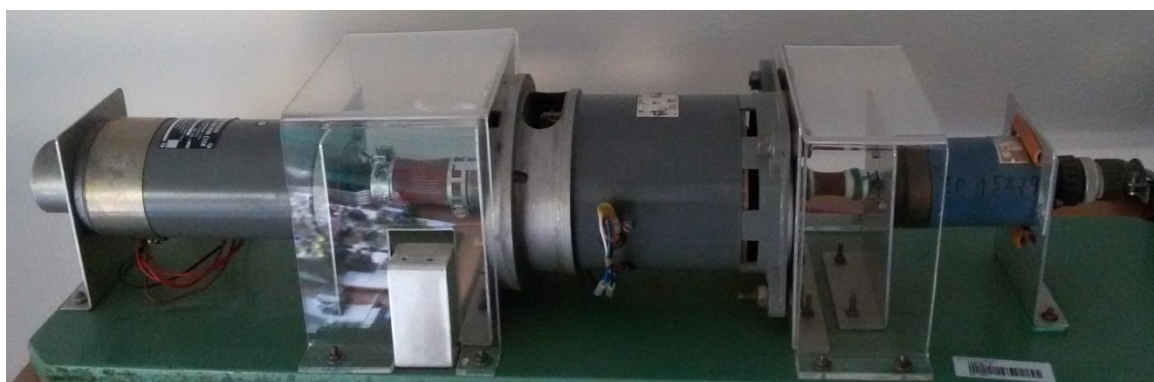
Obr. 2 - Návrh distribuovaného systému

1.2 Stejnoseměrný motor

Tato úloha je zaměřena na řízení otáček stejnosměrného motoru napájeného pulsním měničem. Otáčky motoru jsou snímány impulsním snímačem IRC 120/1250 poskytující 1250 impulsů na otáčku. Otáčky motoru jsou ovládány pulzně šířkovou modulací.

Jednotka obsahující řídicí obvody disponuje vstupy pro ovládání pomocí PWM a ovládání směru. Na výstupu pak je signál informující o směru otáčení motoru a signál s pulzy přepočítanými z výstupu IRC snímače. Napájecí napětí motoru je 42 V. Řídicí algoritmus byl realizován mikropočítačem JR 552, tedy pro potřeby této úlohy je třeba vytvořit řídicí algoritmy pro PLC.

Tuto úlohu je možné provozovat ve dvou režimech. Jedním je režim ovládání bez zpětné vazby, kdy je akční veličina nastavena přímo v ovládací a monitorovací aplikaci. Druhým je režim regulace, kdy je akční veličina počítána pomocí regulátoru. [Jančík, L., 1993]



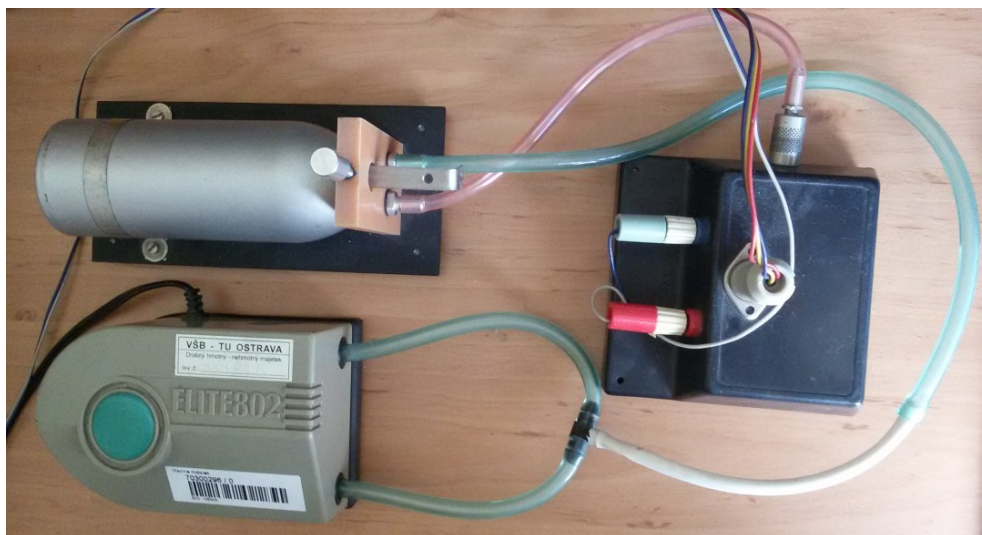
Obr. 3 - Stejnosměrný motor

K napájení

1.3 Tlakovzdušná soustava

Tato úloha realizuje řízení tlakovzdušné soustavy. Zdrojem tlaku je motorek ELITE 802. Řízení velikosti výstupního tlaku vzduchu je dosahováno změnou napájecího napětí v tomto případě 0 – 10 V. Snímač tlaku vzduchu ve vzdušníku pro realizaci zpětné vazby je TMK 518S s rozsahem 0 – 1700 Pa.

Tuto úlohu je možné rovněž provozovat ve dvou režimech. Tedy v režimu ovládání bez zpětné vazby a v režimu regulace, kdy je akční veličina počítána pomocí regulátoru. Řídicí algoritmy byly realizovány pomocí mikropočítače BAST 552. Takže i v tomto případě je nutné vytvořit nové algoritmy pro řízení pomocí PLC. [Jančík, L., 1993]



Obr. 4 - Tlakovzdušná soustava

1.4 Model výtahu

V této úloze je realizováno řízení pětipatrového modelu výtahu pomocí vizualizačního software Control Web 6. Samotné řídicí algoritmy této úlohy jsou již naprogramovány pro použitá PLC firmy ABB. Tedy v rámci této práce není zapotřebí tvorby nových algoritmů. Pouze je třeba naprogramovat komunikaci přes Modbus protokol. Pohyb výtahu je realizován použitím krokového motorku.

Výtah lze ovládat buď přes aplikaci vytvořenou v prostředí Control Web 6, nebo přímo na ovládacím panelu úlohy. Na tomto panelu jsou tlačítka, která slouží k ovládání výtahu. Jedna část z těchto tlačítek slouží k přivolání výtahu v aktuálním patře. Druhá část tlačítek pak slouží k ovládání výtahu zevnitř kabiny. Rozpoznávání patra, ve kterém se výtah aktuálně nachází, je realizováno pomocí jazýčkových relé. Tato relé se spínají pomocí magnetu, který je připevněn k výtahu. Kabina výtahu je připevněna ke krokovému motorku pomocí provázku.



Obr. 5 - Model výtahu

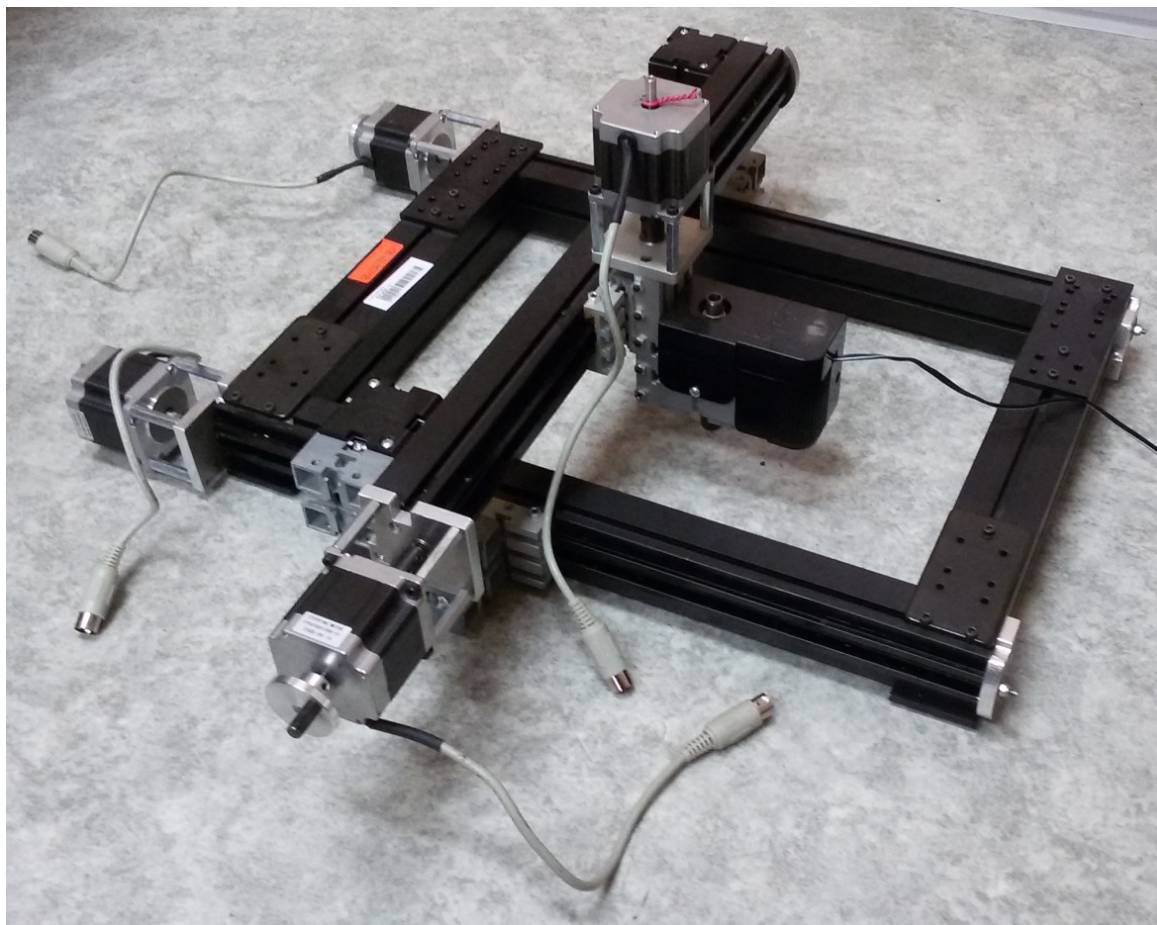
Krokový motorek pohánějící model výtahu je od firmy SANYO DENKI typu 103 - H548 - 04500. Tento krokový motorek má 4 cívký, které jsou napájeny z výkonového členu. Úhel pootočení při vykonání jednoho kroku je $1,8^\circ$. Tedy pro vykonání jedné otáčky je nutné provést 200 kroků. [Jančík, L., 1993]

1.5 Vrtání plošných spojů

Tato úloha představuje realizaci vrtání plošných spojů navržených v softwaru Eagle na portálové frézce. Portálová frézka je poskládaná z modulů firmy Cool Tool, z nichž se dá poskládat nespočet CNC zařízení. Polohování frézy je řešeno pomocí krokových motorků.

Krokové motorky této frézky jsou připojeny k šroubovým tyčím, jež následně pohybují s přemísťovanými objekty vsazenými do saní. Stejně jako u modelu výtahu, i zde se krokový motorek při jednom kroku pootočí o úhel $1,8^\circ$. Závity šroubových tyčí mají

stoupání 1 mm. Tím pádem za jednu otáčku krokového motorku, tedy 200 kroků, se přemísťovaný objekt posune o 1 mm.



Obr. 6 - Portálová frézka

Napájení krokových motorků je prováděno pomocí spínacího výkonového obvodu. Princip tohoto obvodu je následující. Když je z PLC přiveden elektrický proud do tranzistoru, tranzistor se otevře a spojí se výkonový obvod obsahující příslušnou cívku krokového motorku.



Obr. 7 - Výkonový obvod

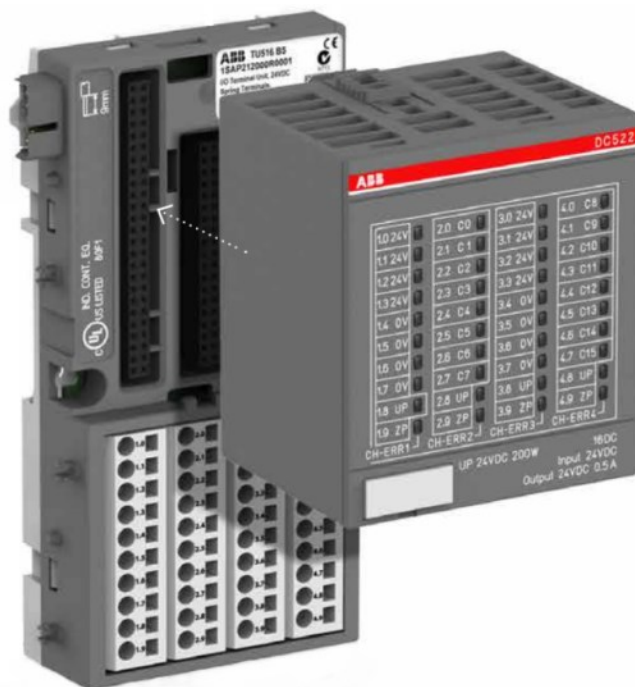
1.6 Použitá PLC

Použitá byla tři PLC řady AC500 od firmy ABB viz obr. 3. Tato řada je na trhu již několik let a stále se přidávají nové modely, které splňují požadavky dnešní doby. V nabídce firmy je velké množství rozšiřovacích modulů, takže v závislosti na konkrétních požadavcích lze sestavit různé kombinace PLC.



Obr. 8 - Použité PLC firmy ABB

Konkrétní typ použitého PLC je PM571-ETH. K němu jsou připojeny analogový modul AX521, digitální modul DC523 a komunikační modul CM572-DP podporující komunikaci přes protokol PROFIBUS.



Obr. 9 - I/O modul pro PLC firmy ABB [ABB, 2014]

Modul PM571-ETH

PM571-ETH je CPU modul, k němuž je možno připojovat další rozšiřující moduly. [ABB, 2006]

Parametry:

Konektivita: Ethernet, 2x RS232/RS485, FBP

Komunikační protokoly: Modbus RTU
Modbus TCP
UDP/IP
CS31 Master
ASCII

Délka 1 cyklu (minimum):	1000 instrukcí (boolean)	0.05 ms
	1000 instrukcí (word)	0.05 ms
	1000 instrukcí (plovoucí desetinná čárka)	0.5 ms

Modul AX521

Modul AX521 obsahuje 4 analogové vstupy a 4 analogové výstupy.

Parametry:

- 4 konfigurovatelné analogové vstupy: 0...10 V
-10...10 V
0...20 mA
4...20 mA
Pt100
Pt1000
Ni1000
- 4 konfigurovatelné analogové výstupy: -10...10 V (max. ± 10 mA)
0...20 mA (odpor zátěže 0...500 Ω)
4...20 mA (odpor zátěže 0...500 Ω)

Modul DC523

Modul DC523 obsahuje 24 konfigurovatelných pinů, které můžou sloužit jako digitální vstupy i výstupy a 8 pinů, které mohou posloužit k napájení senzorů. Tři tyto piny lze rovněž nakonfigurovat jako rychlý čítač pulzů až do 50 kHz. [ABB, 2014]

Parametry:

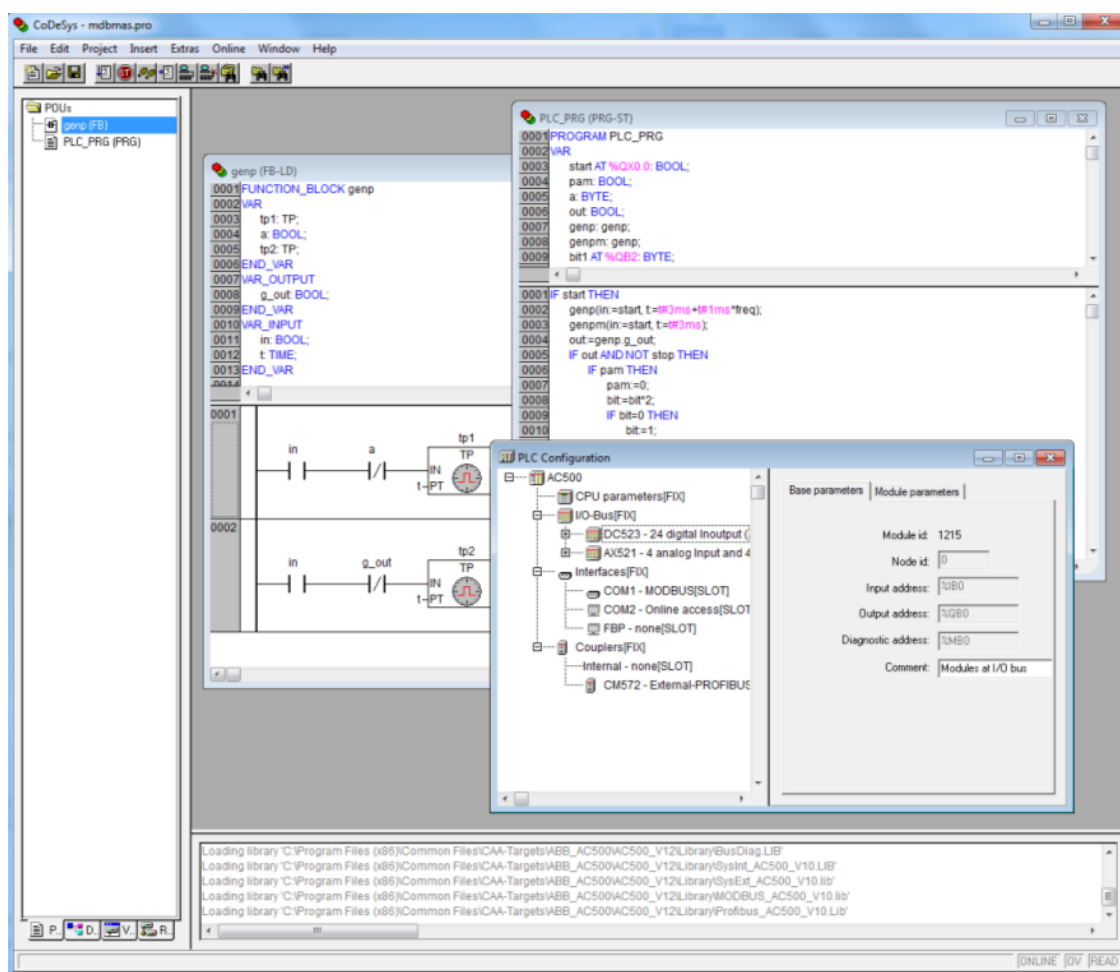
- Maximální výstupní proud: 0,5 A
Maximální zátěž celého modulu: 300 W

2 CODESYS

Pro programování svých PLC poskytuje firma ABB software firmy 3S-Smart Software Solutions zvaný CODESYS (COntroller DEvelopment SYStem). CODESYS je rozsáhlý softwarový nástroj pro průmyslové automatizační technologie. [Vojáček, A., 2011]

2.1 Programovací prostředí a komunikace s PLC

Prostředí programu CODESYS se skládá z několika částí, jak je vidět na obr. 5. Hlavní pracovní část prostředí je určená pro okna s programy, pro okna s jednotlivými dílčími funkcemi a funkčními bloky a pro okna s různými nastaveními. Nahoře je nástrojová lišta, ve které jsou ikony s nástroji pro zovna aktivní část vytvářeného programu a nástroje pro jeho ovládání. Vlevo jsou karty s nabídkami programových organizačních jednotek, uživatelem vytvořených datových typů, vizualizací a s nabídkami různých nastavení PLC, komunikace, knihoven apod. Dole je pak zápis provedených akcí, případně chyb, které byly objeveny při kompilaci napsaného programu.



Obr. 10 - Prostředí programu CODESYS

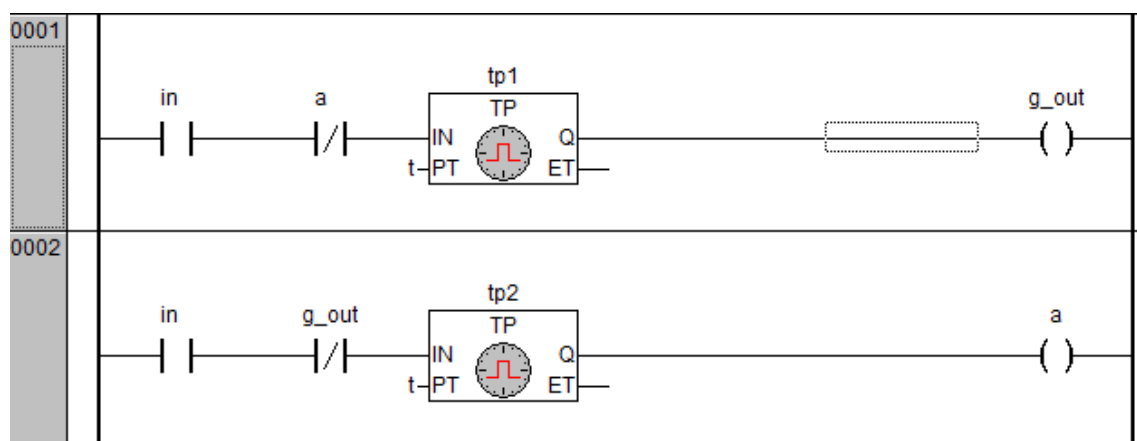
Propojení na softwarové úrovni se provádí v programu CODESYS. Pomocí nástroje IP Config musí být k PLC přiřazena IP adresa, pomocí které pak s PC komunikuje. Po nakonfigurování použitého PLC a jeho modulů musí být v nástroji SYCON.net nastaveno konkrétní spojení programu s PLC. Je tedy třeba u ethernetového připojení napsat do komunikačních parametrů IP adresu onoho. Stejná adresa pak musí být zapsána i v položce Communication Parameters z menu Options.

2.2 Programovací jazyky

V prostředí CODESYS se dají programy psát ve více jazycích podle normy IEC 61131-3. Každá část programu může být napsána v jiném jazyce. Dokonce je možné jednotlivé části programu převádět z jednoho jazyka do druhého. Nicméně musí to být konverze z vyššího jazyka na nižší. [Vojáček, A., 2011]

Liniové schéma

Liniové schéma je grafický jazyk, také nazýván jako jazyk kontaktních schémat. Je založen na grafické podobě reléové logiky. Je organizován pomocí sítě propojených grafických prvků. Tato síť je zleva i zprava ohraničená dvěma svislými čarami, nazývanými napájecí sběrnice. Mezi sběrnicemi je tzv. příčka, která může být rozvětvená. Každý úsek této příčky může být ve stavu TRUE nebo FALSE. Do příček mohou být vloženy kontakty, funkce a funkční bloky.

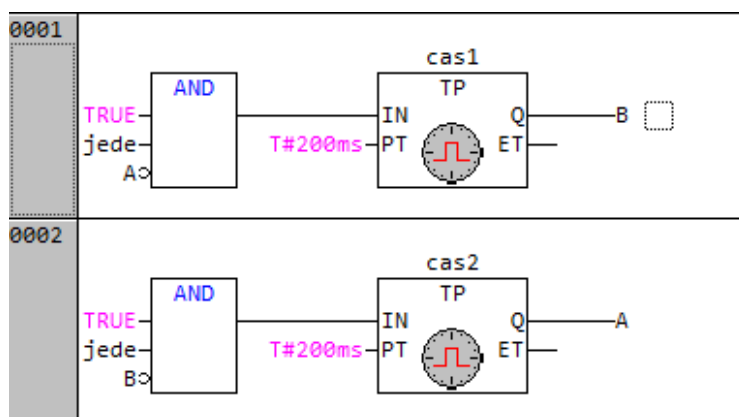


Obr. 11 – Liniové schéma

Schéma funkčních bloků

Druhý grafický jazyk schéma funkčních bloků vyjadřuje pomocí vzájemně propojených grafických bloků chování funkcí, funkčních bloků a programů. Jde

o systém prvků, které zpracovávají signály. Často se zde používají standartní funkční bloky jako např. prvky pro detekci náběžné a sestupné hrany, čítače, časovače apod. Podobně jako u liniového schématu je schéma funkčních bloků uspořádáno do jednotlivých řádků.



Obr. 12 - Funkční blokové schéma

Instrukční sada

Textový jazyk IL (Instruction List) je tvořen sekvencí instrukcí, z nichž každá začíná na novém řádku. Každá instrukce obsahuje operátor a v závislosti na typu operace jeden nebo více operandů oddělených čárkou. Před instrukcemi může být návěští zakončeno dvojtečkou. Na konci řádku u každé instrukce může být komentář.

0001	LD	TRUE
0002	AND	jede
0003	ST	gener.jede
0004	CAL	gener
0005		
0006	LD	gener.A
0007	ST	pulsy

Obr. 13 - Seznam instrukcí

Jazyk strukturovaného textu

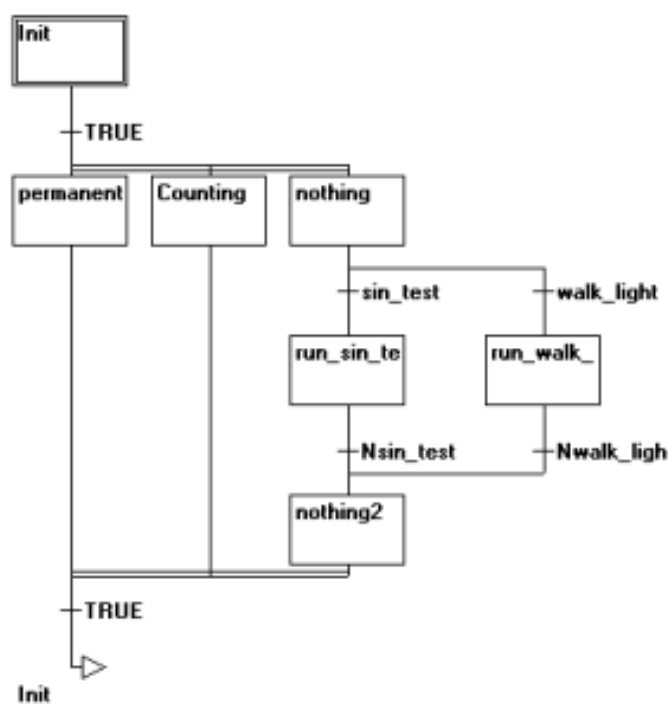
Textový jazyk ST (Structured Text) je výkonný vyšší programovací jazyk, jehož syntaxe je dána povolenými výrazy a příkazy. Vyhodnocením výrazu vyjde hodnota v některém z definovaných datových typů. Na jednom řádku může být více příkazů oddělených středníkem. Tento jazyk je vhodný k tvorbě komplexních funkčních bloků, které pak mohou být použity v jiném programovacím jazyku.

0013	IF vystup>8 THEN
0014	vystup:=1;
0015	ok:=1;
0016	ELSE
0017	ok:=0;
0018	END_IF

Obr. 14 - Strukturovaný text

Sekvenční funkční diagram

Další grafický jazyk SFC (Sequential Function Chart) popisuje sekvenční chování programu. Skládá se z kroků a přechodů zobrazených v síti. Každý krok reprezentuje stav řízeného systému a je k němu přiřazen blok akcí. Přechod pak určuje podmínky, jež musí být splněny, aby mohl být aktivován krok, který přechodu předchází.



Obr. 15 – Sekvenční funkční diagram [Vojáček, A., 2011]

3 Control Web 6

Control Web 6 je univerzální SCADA/HMI nástroj. Jeho komponentová architektura je objektově-orientovaná a zajišťuje aplikacím systému Control Web široký rozsah nasazení od jednoduchých časově nenáročných vizualizací až po komplexní řídicí aplikace reálného času.

Control Web 6 vychází z osvědčené architektury jeho předchůdců Control Panel a starších verzí Control Web. Nasazení těchto systémů od jaderných elektráren a celopodnikových vizualizačních systémů až po přímé řízení strojů a jednoduché vizualizace vypovídá o velmi širokých možnostech této architektury. Novější verze tohoto softwaru jsou zpětně kompatibilní s aplikacemi vytvořenými ve starších verzích. Tedy chrání investice do již existujících aplikací, vzdělání a know-how. [Moravské přístroje]

3.1 Vývojové prostředí

Software Control Web 6 podporuje takzvané dvojcestné programování. Znamená to, že aplikace lze vytvářet najednou jak v grafickém prostředí, tak i v textovém editoru. Přechodu mezi jednotlivými režimy vývoje aplikace se říká překlápění. Tato dualita umožňuje podle potřeby vybrat momentálně nejlepší postup tvorby aplikace.

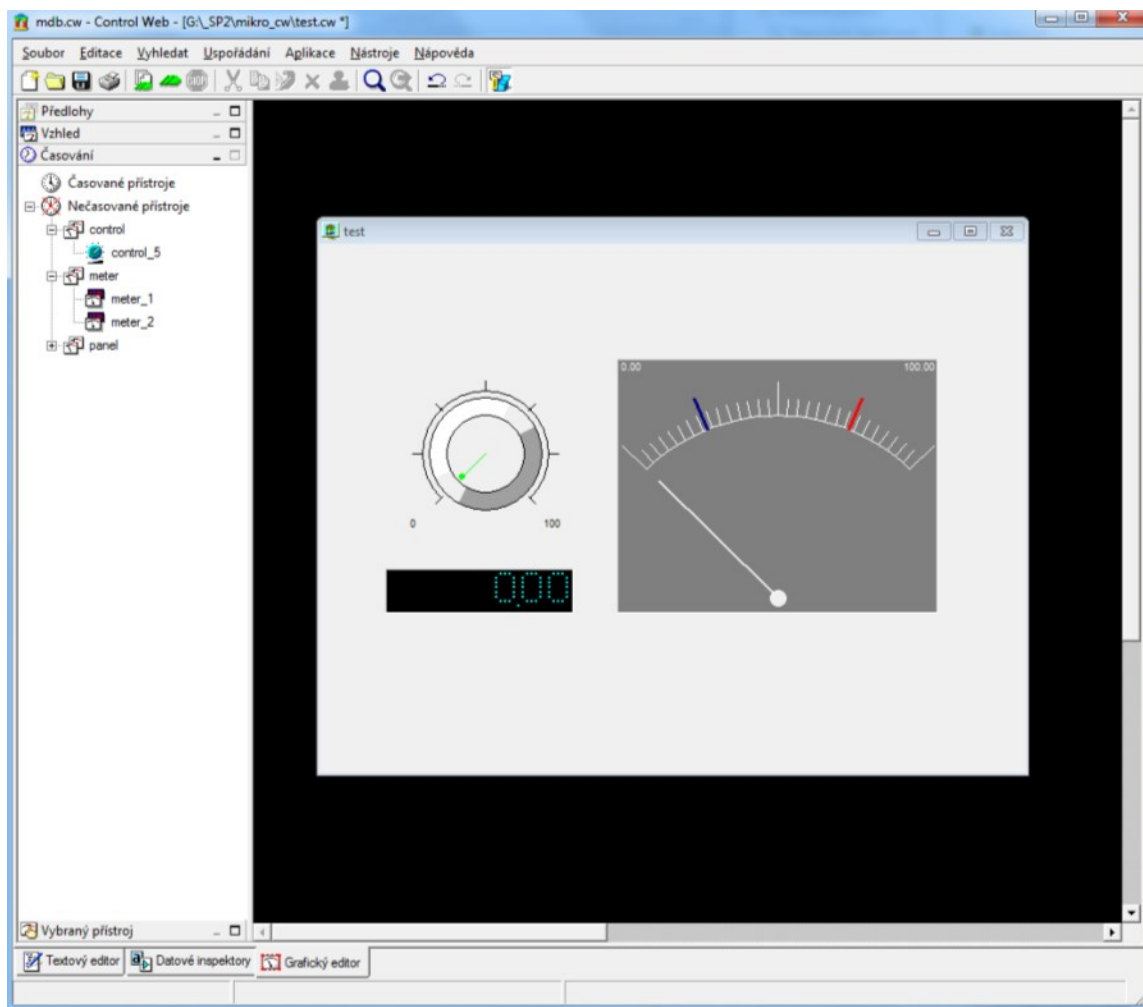
Pro ukládání a zálohování však slouží textový editor. A tak, i když jsou oba způsoby tvorby softwaru zástupné, textový editor má vyšší prioritu. Znamená to, že veškeré grafické prvky, jejich parametry, procedury apod. mají přesně daný zápis, který je nutno při psaní kódu bezpodmínečně dodržet jinak Control Web oznámí chybu. Mechanismus, který rozhoduje, zda daný text vyhovuje systému, se nazývá překlad. Generování pak provádí přesný opak než překlad. Tedy když se jakkoliv změní aplikace v grafickém prostředí, Control Web pro tuto změnu vygeneruje příslušný kód.



Obr. 16 - Překlad a generování

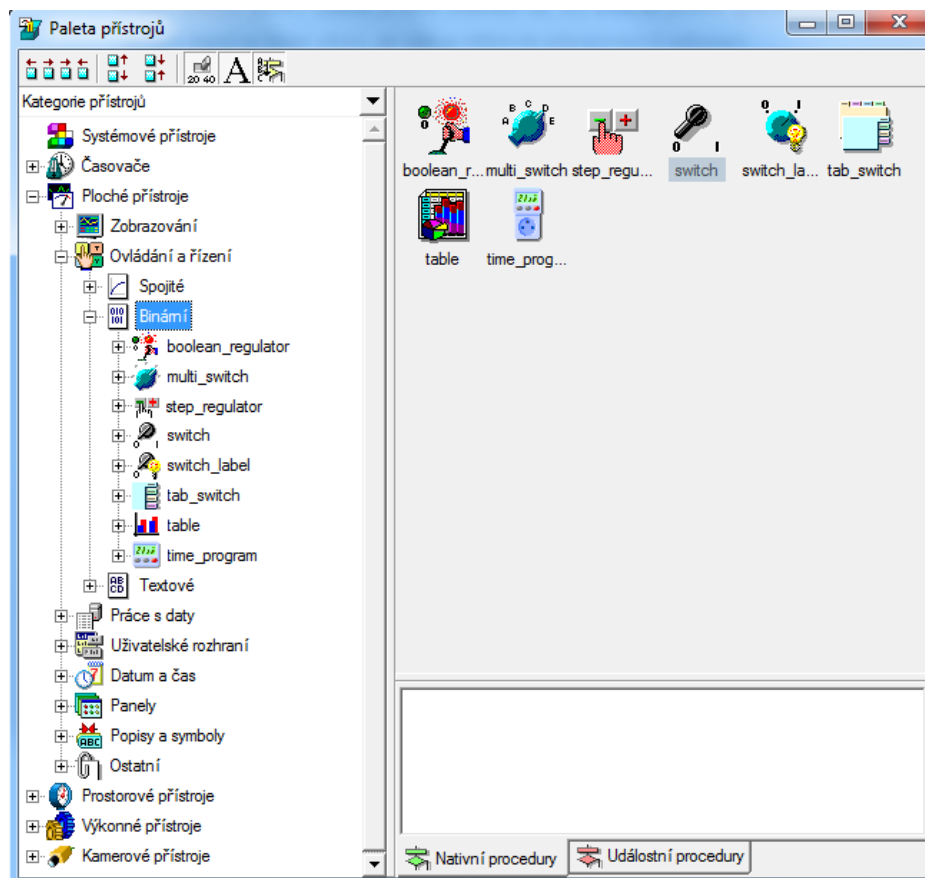
3.2 Grafický editor

Grafický editor poskytuje vizuální pohled na tvořenou aplikaci. Lze do něj podle potřeby vkládat jednotlivé přístroje, upravovat jejich vzhled, navazovat na ně proměnné a mnoho dalších operací.



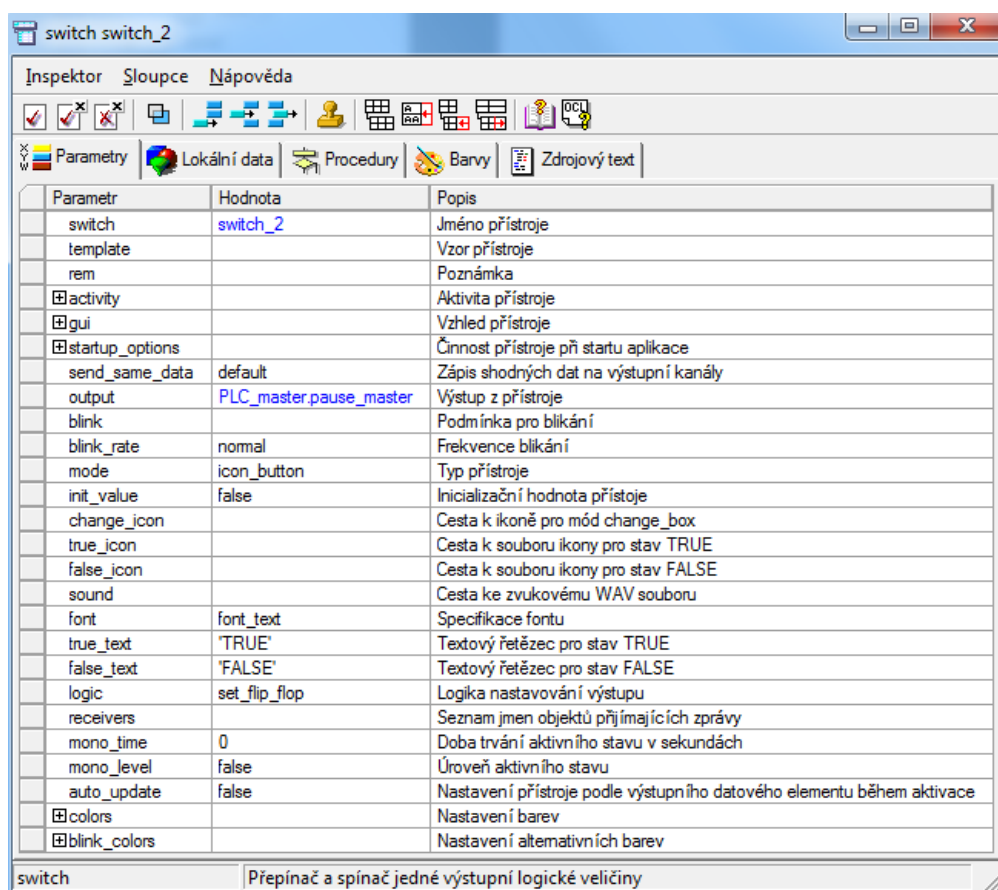
Obr. 17 - Grafický editor

Pro vkládání jednotlivých grafických prvků slouží paleta přístrojů. Ta obsahuje strom s kategoriemi a podkategoriemi přístrojů. Každý jednotlivý přístroj lze do aplikace přetáhnout myší. Vzhled přístroje, jak je zobrazen v paletě přístrojů, však není jediný možný. Pomocí přiloženého nástroje Ikoner lze spouště přístrojů dotvářet vlastní vzhled, čímž se možnosti grafického provedení aplikace stávají takřka neomezené. [Moravské přístroje]



Obr. 18 - Paleta přístrojů

Vzhled a chování vložených přístrojů je možné přizpůsobovat v takzvaném inspektoru přístroje. Jeho funkce jsou rozděleny do samostatných záložek. Zřejmě nejčastěji používaná záložka jsou Parametry. V ní jsou ve stromové struktuře zobrazeny vlastnosti jednotlivých přístrojů, které lze jednoduše nastavit. Důležitou záložkou jsou také Procedury, kde se nachází události, které můžou u daného přístroje nastat, např. jeho aktivace nebo kliknutí myši na přístroj. V těle těchto procedur se pak píše kód, kterým se definuje, co chceme, aby se při dané události stalo. V záložce Textový editor je pak možné vidět výřez celkového kódu, který je přístrojem generován. Ten můžeme libovolně upravovat. Avšak nevýhodou je, že v této záložce nejsou v nabídce žádné vlastnosti přístroje ani procedury. Ty tedy musíme buď znát, anebo je nastavit v předchozích nabídkách a v textovém editoru je pak upravit podle potřeby. [Moravské přístroje]

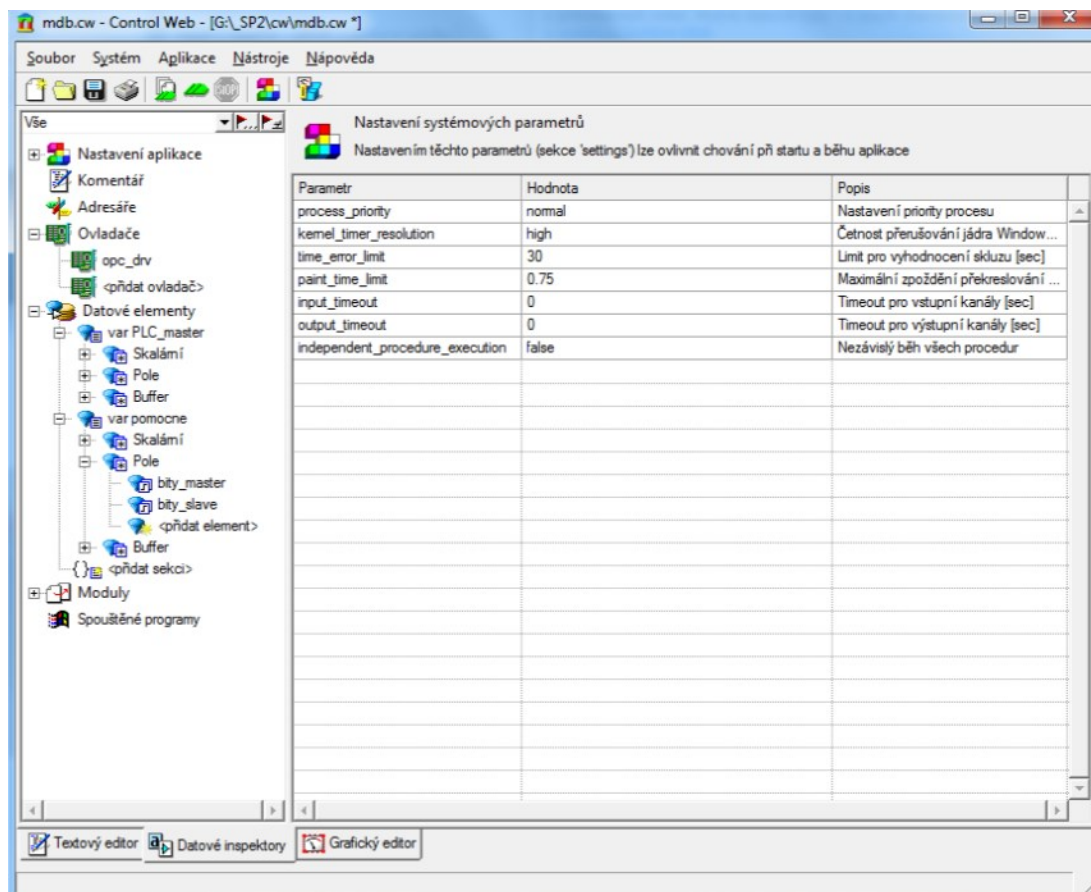


Obr. 19 - Inspektor přístroje

3.3 Datové inspektory

Ve vývojovém prostředí softwaru Control Web 6 je mezi záložkami Textový a Grafický editor třetí záložka, a tou jsou Datové inspektory. Technicky jsou datové inspektory součástí grafického režimu, protože je rovněž lze nahradit kódem v textovém editoru. A naopak, když se v Datových inspektorech něco vytvoří, tak je opět generován do textového editoru příslušný kód.

Datové inspektory zprostředkovávají komunikaci s okolním světem, jsou v nich obsaženy data, vnitřní údaje a nastavení parametrů aplikace. Je zde tedy možné například nadefinovat ovladač, přes který chceme s okolím komunikovat, vytvořit komunikační kanály obsahující potřebné proměnné a vytvořit pomocné proměnné. Proměnné, které se v datových inspektorech vytvoří, jsou globální. Lze k nim tedy přistupovat z libovolného místa programu. [Moravské přístroje]



Obr. 20 - Datové inspektory

3.4 Časování

Systém Control Web 6 je systém reálného času. Této možnosti je možné se vzdát, a tedy program může pracovat jako řízená data. To přináší zjednodušení návrhu a menší nutnost detailně celou úlohu promýšlet. Nicméně je tím ovlivněna možnost systém přesně ovládat.

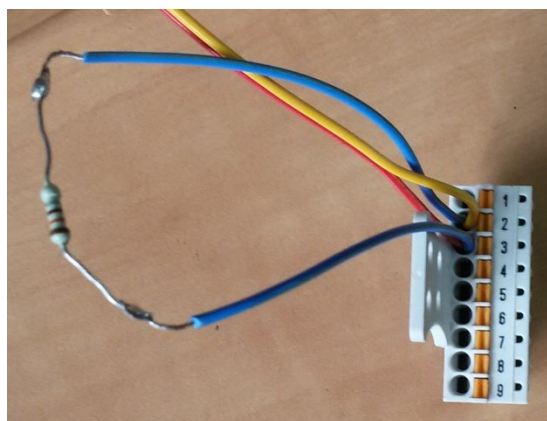
Časování aplikace je pak možné tvořit pomocí přístroje Selector, kde se cyklicky kontrolují podmínky, které určují, kdy se má spustit určitá část skriptu, ovládajícího grafickou vizualizaci aplikace. Dalším velice výkonnou možností jak dosáhnout přesného chodu aplikace je použití příkazu send, kterým je přesně definováno, který přístroj má v daném okamžiku vykonat aktivitu. [Moravské přístroje]

4 Komunikace

Komunikace mezi jednotlivými PLC je realizována pomocí protokolu Modbus RTU. Komunikace PLC s aplikací vytvořenou v prostředí Control Web 6 je pak řešena pomocí OPC protokolu.

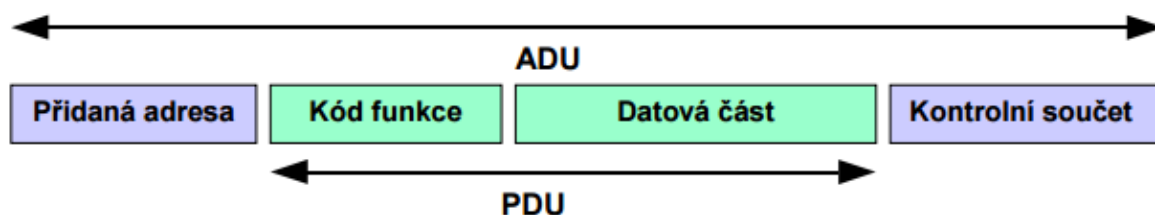
4.1 Modbus RTU

Modbus je protokol typu master/slave. Přídomek RTU v tomto případě znamená, že se jedná o komunikaci přes sériovou linku. Konkrétně byl použit standard RS485. Přes RS485 může komunikovat až 32 zařízení. Komunikace probíhá přes kroucenou dvoulinku, kde musí být na koncových zařízeních zakončovací 120 Ω odpory. Ty slouží k tomu, aby se signál neodrážel zpět, a zároveň zaručují větší odolnost linky vůči rušení.



Obr. 21 - Konektor pro sériovou komunikaci pro PLC ABB AC500

Struktura zprávy protokolu Modbus je na úrovni protokolu PDU (Protocol Data Unit) a je nezávislá na typu komunikační vrstvy. Podle toho na jakém typu síti je Modbus použit, je PDU struktura rozšířena o další části, čímž tvoří zprávu na aplikační úrovni ADU (Application Data Unit). [Ronešová, 2005]



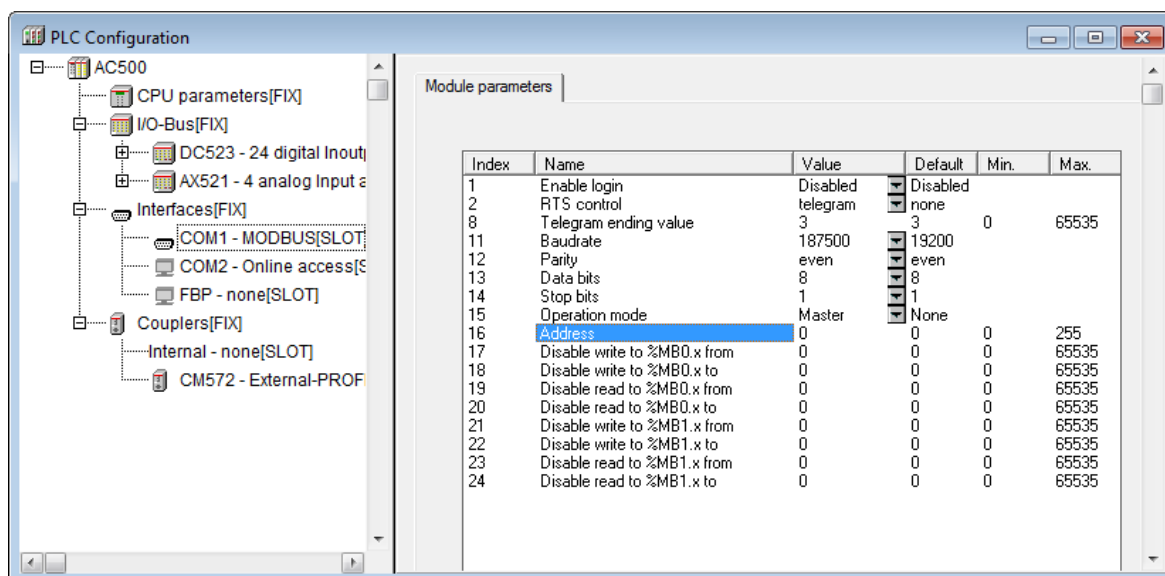
Obr. 22 - Základní tvar Modbus zprávy[Ronešová, 2005]

Přidaná adresa ve zprávě udává, se kterým zařízením chce master komunikovat. Kód funkce pak specifikuje, o jaký typ přenosu dat se bude jednat. Datová část může mít u PLC od ABB délku maximálně 250 bytů.

Tab. 1- Kódy funkcí Modbus zpráv

kód funkce	funkce	maximální délka zprávy
01 nebo 02	čti n bitů	2000 bitů
03 nebo 04	čti n wordů	125 wordů / 62 double wordů
5	zapiš 1 bit	1 bit
6	zapiš 1 word	1 word
7	čti 8 bitů	8 bitů
15	zapiš n bitů	1968 bitů
16	zapiš n wordů	123 wordů / 61 double wordů
22	zapiš s maskováním	
23	čti/zapiš více wordů	

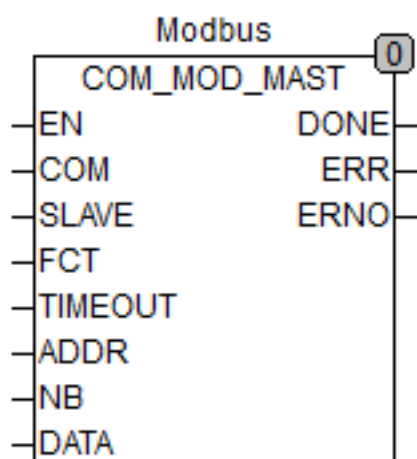
Aby mohlo PLC komunikovat přes Modbus, musí se nejprve v programu CODESYS nakonfigurovat na používaný sériový port. V něm se určí, zdali se jedná o zařízení typu master nebo slave, zapiše námi vybraná adresa zařízení, zvolí rychlost přenosu atd. Pro RS485 musí být RTS řízení nastaveno vždy na telegram



Obr. 23 - Konfigurace Modbus protokolu v programu CODESYS

Pro komunikaci ve vytvořeném kódu se v programu CODESYS používá funkční blok COM_MOD_MAST. V něm se nastavují všechny důležité parametry pro konkrétní komunikaci. Tento funkční blok se vykoná, když do vstupu EN jde signál s náběžnou hranou. Takže pro každou zprávu do něj musí jít tento spouštěcí signál. V této práci se

nachází 4 tyto funkční bloky pro komunikaci se dvěma PLC. Tedy je v softwaru smyčka, která postupně přepíná proměnné umožňující vykonání funkčního bloku. Dále se zde nastavuje, kterým portem na PLC se vyšle zpráva, kód funkce, adresa slave zařízení, množství poslaných dat a adresa prvního poslaného elementu. Z toho důvodu je třeba proměnným sloužícím pro komunikaci přes Modbus přiřazovat pomocí operandu určité konkrétní místo v paměti (např. %MX0.1.1 pro jeden bit). Další data určená pro komunikaci pak musí mít adresu navazující, aby je mohl funkční blok postupně číst a zapisovat.



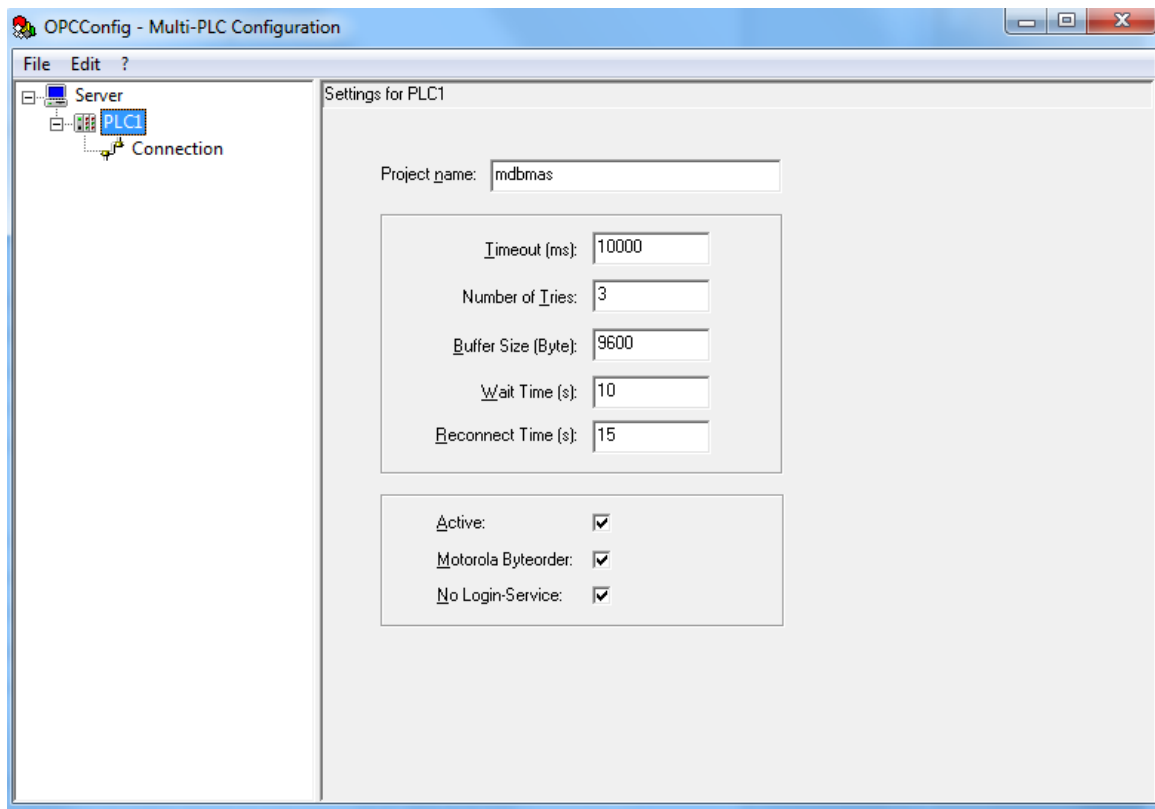
Obr. 24 - Funkční blok ovládající Modbus komunikaci

4.2 OPC komunikace

OLE for Process Control (**OPC**) představuje první úspěšnou iniciativu standardizující komunikační rozhraní mezi různými prvky průmyslové automatizace. Ze standardu **OPC** profitují především uživatelé, kteří díky němu nejsou vázáni na programové a technické vybavení podporující pouze protokoly zavedených firem (zejména pokud tyto firmy brání své postavení právními překážkami zabraňujícími implementaci daného protokolu třetím stranám).

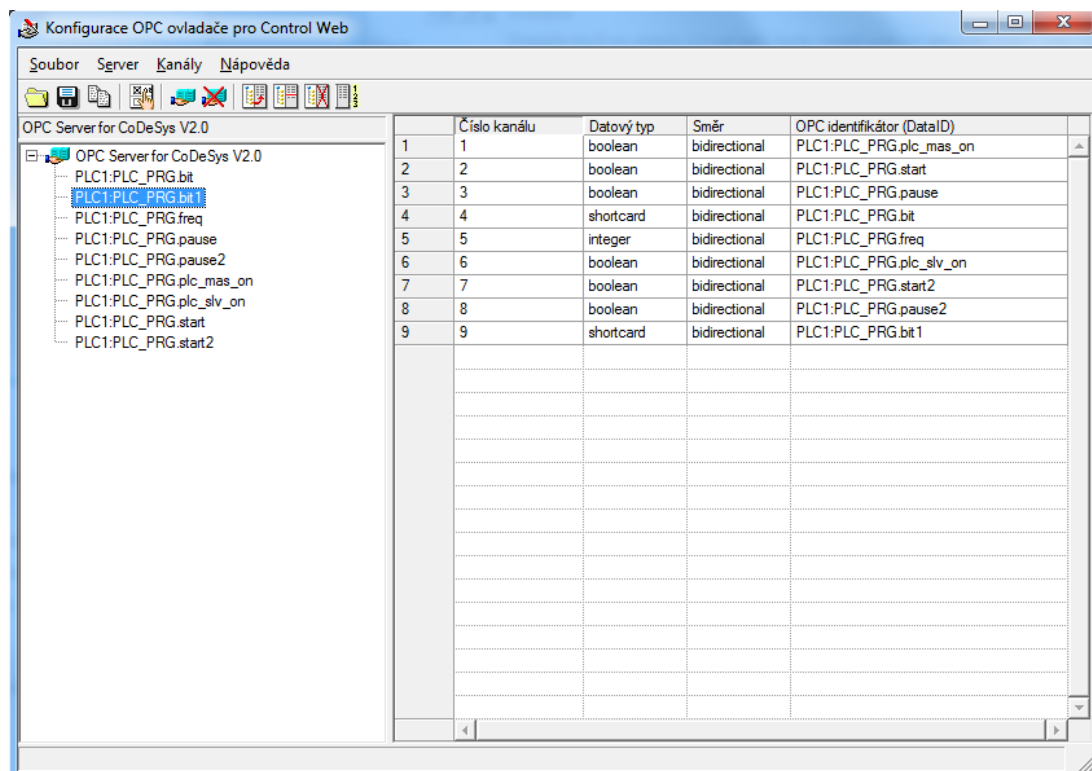
Pro úspěšnou komunikaci je zapotřebí nejprve v programu CODESYS nastavit export potřebných proměnných. To se provádí přes Project -> Options... -> Symbol configuration. Zde je třeba mít zaškrtnutou položku Dump symbol entries. Poté je třeba kliknout na Configure symbol file... a zde si již vybrat proměnné, které chceme exportovat. Při nahrání programu do PLC se pak i vytvoří soubor, se kterým již umí OPC server pracovat.

Dále je třeba nakonfigurovat OPC server. To se provádí v nástroji OPC Configurator. Zde se musí nejprve napsat název projektu, který je nahráný v PLC. Důležité pak je v nastavení připojení zvolit stejné připojení jakým se počítač připojil k PLC. V neposlední řadě se musí i nastavit obnovovací perioda.



Obr. 25 - OPC Configurator

V programu Control Web 6 se pak musí nakonfigurovat OPC ovladač. V případě správného nastavení v OPC Configurator se v konfiguratoru OPC ovladače v programu Control Web načtou proměnné, jež jsou určeny pro OPC komunikaci. Z těchto proměnných si můžeme vybrat ty, které potřebujeme, a přiřadit jim číslo kanálu. Poté se vytvoří soubory *.par a *.dmf, které obsahují informace o těchto proměnných.



Obr. 26 - Konfigurace OPC ovladače v programu Control Web

5 Tvorba aplikací

Úlohy „Stejnoseměrný motor“ a „Tlaková soustava“ nebyly řízeny pomocí PLC tedy pro ně je zapotřebí vytvořit řídicí algoritmy v programu CODESYS. Algoritmy pro úlohu „Vrtání plošných spojů“ sice již byly realizovány pro řízení pomocí PLC, nicméně v této práci je vyvinuta nová aplikace. Jediná úloha kde již nejsou vytvářeny nové algoritmy je úloha „Model výtahu“. Ke všem úlohám však je vytvořena nová ovládací a monitorovací aplikace v prostředí Control Web 6.

5.1 Stejnoseměrný motor

Signály z pohledu PLC pro řízení stejnosměrného motoru jsou následující:

Vstupy: čtení otáček – pulzy z IRC
 čtení směru – 0 V / 5 V
Výstupy: PWM pro ovládání motoru
 ovládání směru – 0 V / 5 V

5.1.1 Vysokorychlostní čítač

Pro zpracování pulzních signálů nabízí modul DC523 vysokorychlostní čítač. Tento čítač je schopný měřit frekvenci pulzů až do 50 kHz. Je možné jej provozovat v několika módech, které je možné nastavit v PLC konfiguraci v prostředí softwaru CODESYS.

- mód 1 - Čítač nahoru.
- mód 2 - Čítač nahoru, nastavení vstupu přes terminál.
- mód 3 - Dva nahoru/dolů čítače.
- mód 4 - Dva nahoru/dolů čítače, jeden s invertovaným vstupem.
- mód 5 - Čítač nahoru/dolů s dynamickým nastavením vstupu přes terminál reagující na vzestupnou hranu.
- mód 6 - Čítač nahoru/dolů s dynamickým nastavením vstupu přes terminál reagující na sestupnou hranu.
- mód 7 - Čítač nahoru/dolů pro čidla polohy.
- mód 8 - Reservován.
- mód 9 - Čítač nahoru/dolů pro čidla polohy, pulsy zdvojnásobené.
- mód 10 - Čítač nahoru/dolů pro čidla polohy, pulsy zčtyřnásobené.

V této práci byl použit mód 1. Nicméně procesorová jednotka zřejmě čte data z vysokorychlostního čítače v intervalu delším, než je frekvence čítání. A tedy dochází k nerovnoměrnému čtení množství přijatých pulzů a vyhodnocený signál je pak velmi zašuměný. Z toho důvodu je vhodné navrhnout filtr dolní propust.

5.1.2 Dolní propust

Pro potřeby zpřesnění čtení otáček motoru se ukázalo jako nejlepší použít filtr dolní propust druhého řádu. Experimentálně byl zjištěn nejlepší přenos filtru.

$$G_F = \frac{900}{s^2 + 30s + 900}$$

Takto navržený filtr je pak rekurzivně naprogramován.

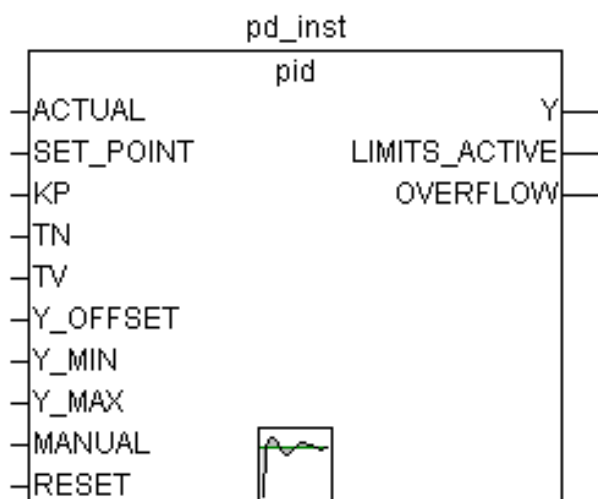
```
y2:=(otacky_read-otacky)*900-y1*30;  
y1:=y1+y2*0.01;  
otacky:=y+y1*0.01;
```

Obr. 27 - Naprogramovaný filtr

5.1.3 Řízení

Moduly PLC firmy ABB dostupné na katedře neobsahují PWM výstupy. Z toho důvodu je nutné transformovat napěťový signál na PWM. Pro tyto účely již byl vytvořen obvod s PIC mikrokontrolérem. Na výstup PLC je tedy zapotřebí pouze přivést napětí 0...5 V. Toho je docíleno pomocí analogového výstupu z modulu AX521. Požadované napětí se generuje v závislosti na hodnotě výstupní proměnné typu integer.

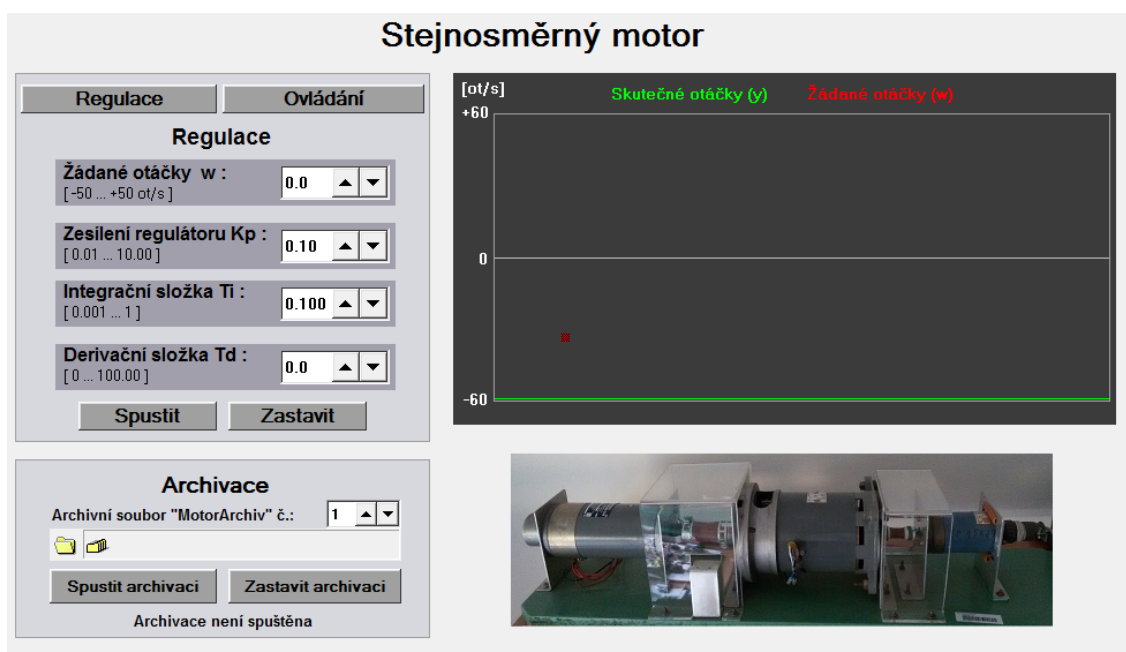
Regulace je v programu CODESYS řešena pomocí funkčního bloku, který rekurzivně vypočítává podle zadaných parametrů výstupní veličinu.



Obr. 28 - Blok pro PID regulaci

5.1.4 Ovládací a monitorovací aplikace

Aplikace v prostředí Control Web 6 umožňuje přepínání mezi ovládáním a regulací. Průběhy jednotlivých veličin jsou zobrazovány v grafu. Pomocí přístroje archiver je umožněna archivace naměřených dat



Obr. 29 - Úloha „Stejnoseměrný motor“ v prostředí Control Web 6

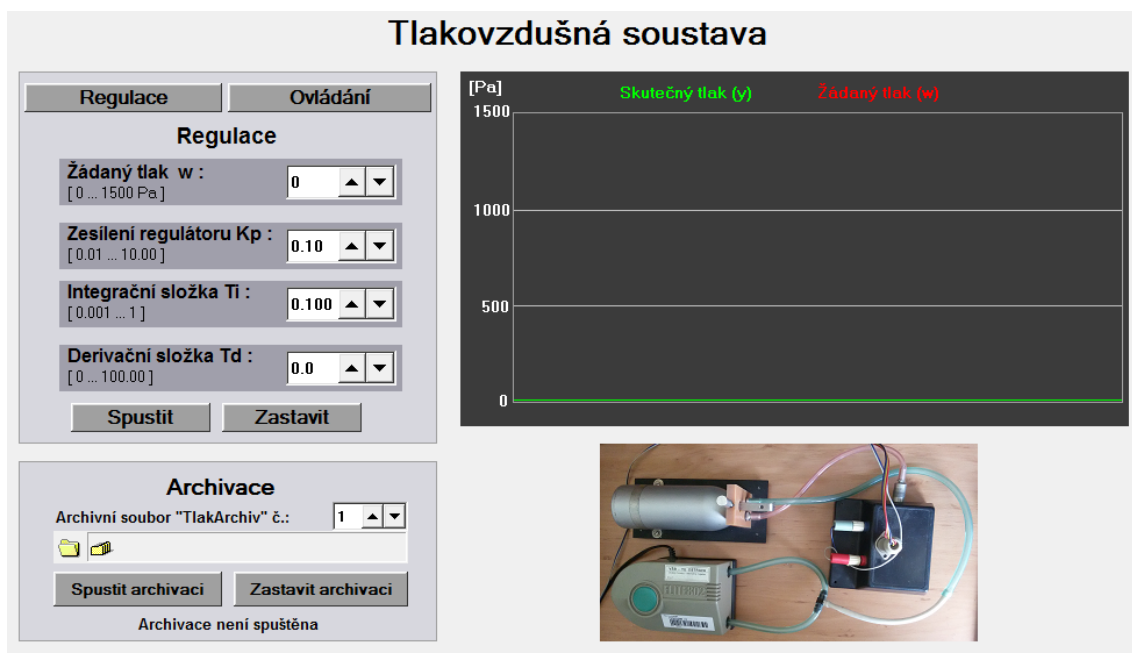
5.2 Tlakovzdušná soustava

Signály z pohledu PLC pro řízení tlakovzdušné soustavy jsou následující:

Vstupy: čtení velikosti tlaku – 0 V / 10 V

Výstupy: ovládání napětí zdroje tlaku – 0 V / 10 V

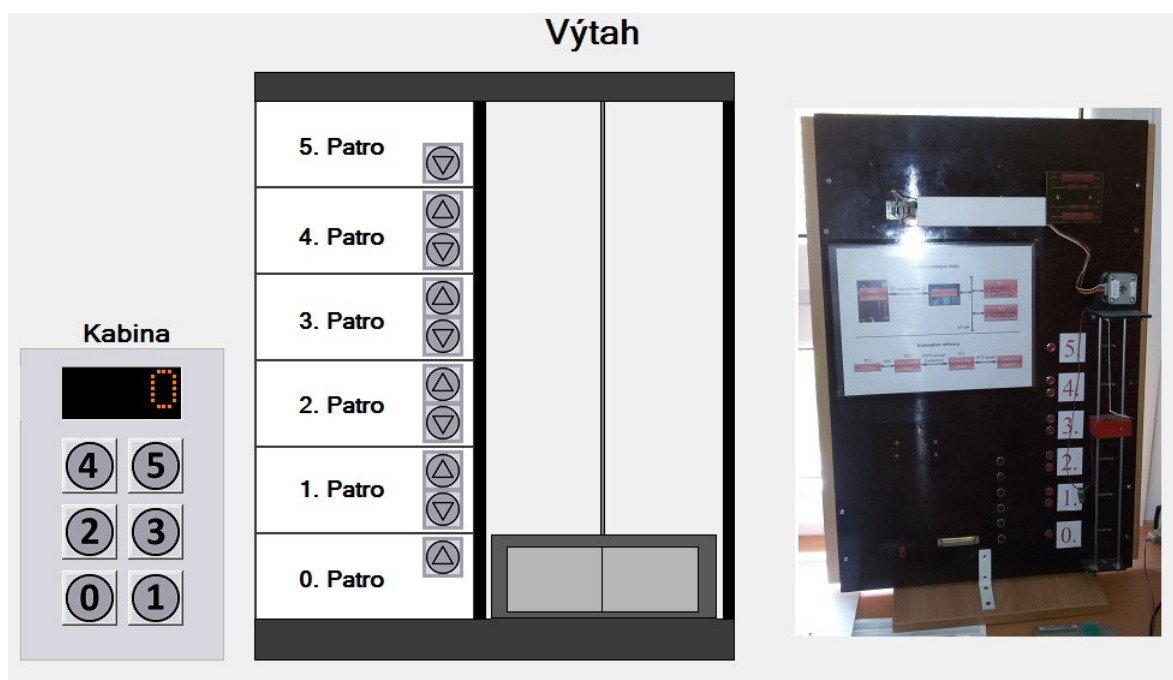
Smysl této úlohy je stejný, jako tomu je u úlohy „Stejnoseměrný motor“. Takže i naprogramování aplikace je velice podobné. Rozdíl je akorát ve velikosti vstupního a výstupního napětí a absence komplikací v podobě PWM vstupu a výstupu. Rovněž část aplikace v prostředí Control Web 6 je principiálně totožná.



Obr. 30 - Úloha „Tlakovzdušná soustava“ v prostředí Control Web 6

5.3 Model výťahu

U této úlohy je použita již hotová aplikace pro PLC. Je tedy doplněna jen o program zprostředkovávající komunikaci přes Modbus RTU. Najednou může probíhat komunikace pouze s jedním PLC. Takže program obsahuje postupně spouštěné funkční bloky, ve kterých se buď čtou, nebo zapisují data potřebné pro řízení příslušných úloh.



Obr. 31 - Úloha „Model výtahu“ v prostředí Control Web 6

5.4 Vrtání plošných spojů

Ačkoliv tato úloha byla již dříve vypracována, jsou v této práci vytvořeny nové algoritmy pro řešení vrtání plošných spojů na portálové frézce.

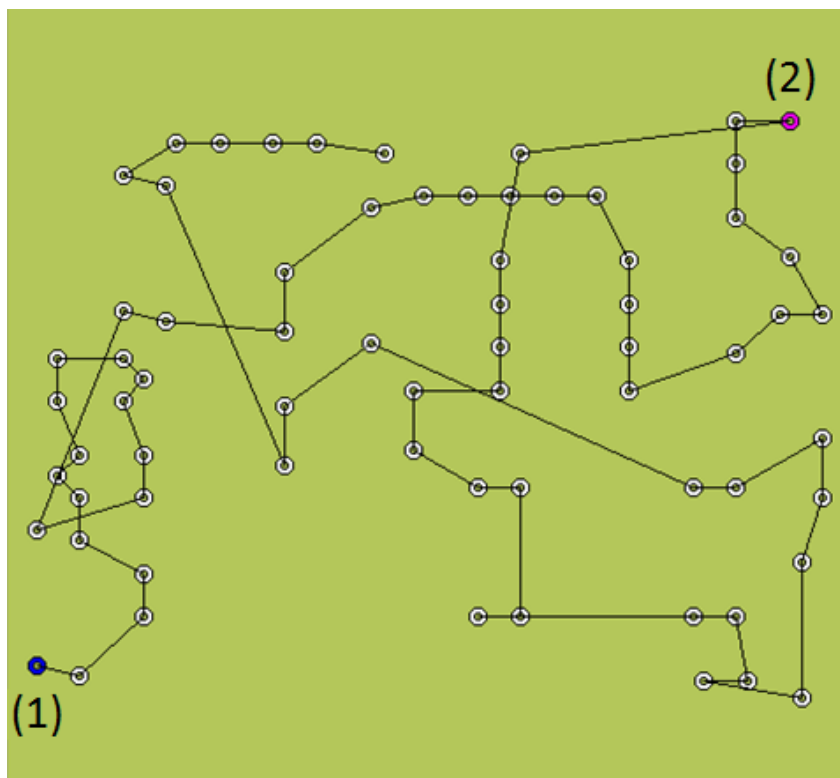
5.4.1 Ovládací a monitorovací aplikace

Nejprve je třeba do aplikace v prostředí Control Web 6 načíst *.ref soubor obsahující informace o umístění děr na plošném spoji. Tento soubor se musí vyexportovat z programu Eagle ve formátu Unidat. S takovýmto souborem pak umí Control Web 6 pracovat jako s textovým souborem a je tedy možné vytvořit seznam děr.

Je třeba určit trajektorii, kterou musí fréza vykonat, aby byly vyvrtány všechny díry. Principiálně algoritmus pracuje tak, že se jako první díra se zvolí ta, která je nejbližší levému dolnímu rohu. Potom se prozkoumávají souřadnice ostatních děr. Díra, která je pak nejbližší této aktuální díře, je pak přidána do posloupnosti vrtaných děr. Tento postup se pak opakuje s tím, že je ošetřeno, aby nebyly započítávány díry, které už se v posloupnosti nachází.

Aplikace rovněž obsahuje korekci natočení tištěného spoje při vkládání do portálové frézky. K tomu slouží referenční body. První referenční bod je v levém dolním rohu, druhý pak v pravém horním rohu. Na pozici referenčních bodů se pak musí nastavit fréza a

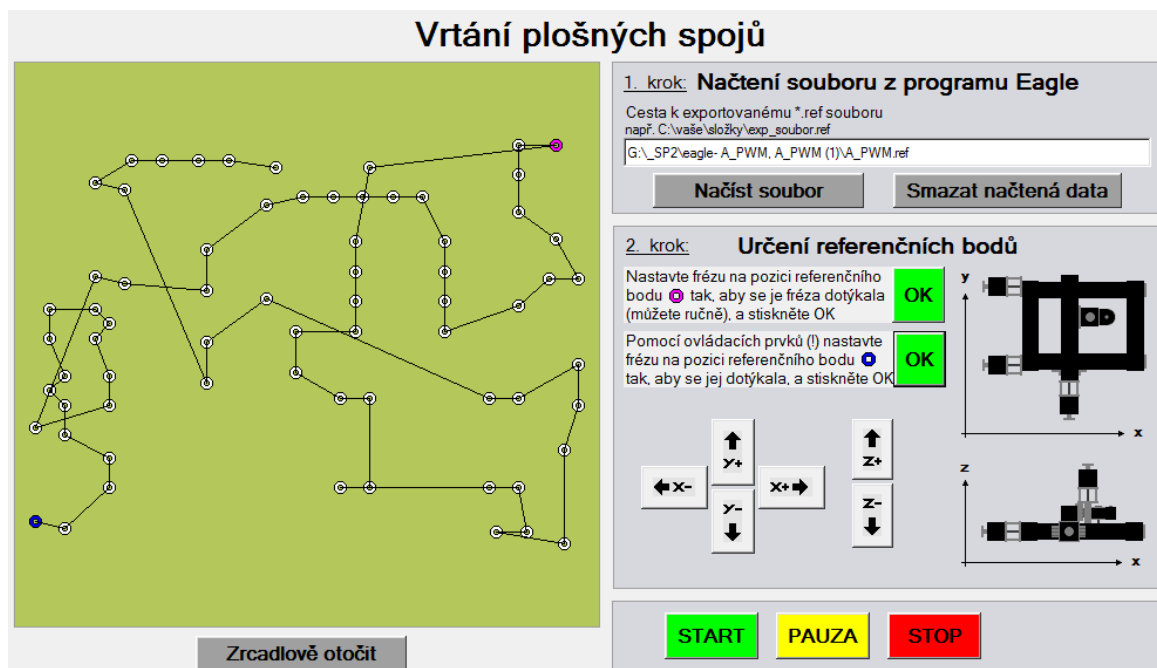
zaznamenají se souřadnice, kterých dosáhla. Na základě těchto údajů se vypočítá úhel pootočení vůči původním souřadnicím. Pomocí tohoto úhlu se pak přepočítají souřadnice všech děr. Plošný spoj je možné pootočit do jakéhokoliv úhlu. Je však nutné, aby byl do frézky vložen tak, aby byla strana spoje, která je zobrazena v aplikaci, byla otočena směrem nahoru. Nicméně aplikace obsahuje možnost zrcadlově otočit plošný spoj. Podle toho se pak i přepočítají souřadnice děr.



Obr. 32 - Referenční body

Poté je možné spustit samotné vrtání děr. Ovládací prvky aplikace jsou vytvořeny tak, aby nebylo možné provádět operace bez příslušných prerekvizit. Tedy například není možné spustit úlohu vrtání děr, pokud nejsou určeny referenční body a vypočítáno pootočení plošného spoje.

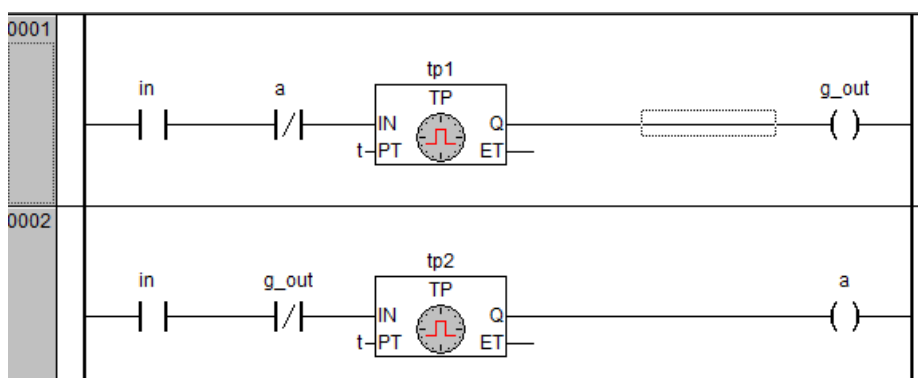
Vrtání pak probíhá tak, že aplikace pošle přes OPC protokol spolu se souřadnicemi díry příkaz k jejímu vyvrtání. Program v PLC pak přesune frézu na zadané souřadnice, díru vyvrtá a pošle zpět informaci o tom, že daná díra byla vyvrtána. Ovládací aplikace poté načte souřadnice další díry, pošle je do PLC a celý proces se takto opakuje až do úspěšného vyvrtání všech děr.



Obr. 33 - Úloha „Vrtání plošných spojů“ v prostředí Control Web 6

5.4.2 Řízení krokových motorků

Pohyb krokového motorku je zajišťován vstupními pulsy spínajícími jednotlivé cívky. Proto je třeba vytvořit generátor pulsů v podobě astabilního klopného obvodu. TP časovač má při hodnotě vstupu TRUE hodnotu výstupu TRUE, která se po daném čase přepne na FALSE. Při hodnotě výstupu FALSE se pak změní vstupní hodnota před druhým časovačem na TRUE a tímto způsobem se jednotlivé proměnné neustále přepínají. Vyvedením jedné z těchto proměnných na výstup tohoto bloku se pak generují pulsy. Perioda pulsů je pak rovna součtu zadaných časových konstant. Ovládání periody přepínání bitů je pak řešeno přičítáním proměnné k časovým konstantám v hlavní části programu.



Obr. 34 - Generátor pulsů

Algoritmus přepínání výstupních bitů byl jednoduše realizován ve strukturovaném textu tak, že při vstupním impulsu je aktuální hodnota výstupu vynásobena dvěma. Po dosažení maximální hodnoty je ještě zapotřebí přepnout hodnotu výstupu na počátek prováděné smyčky, tedy na hodnotu 1, a zároveň při té příležitosti nastavit v nadřazeném PLC pulz pro PLC podřízené.

Ještě je třeba ošetřit skutečnost, že během jednoho impulsu program provádí operaci vícekrát. To je vyřešeno tak, že se do algoritmu zavede paměťová proměnná. Pak když je hodnota vstupu FALSE, hodnota paměti je TRUE. Po přepnutí vstupu na TRUE se provede operace, jen když je hodnota paměti TRUE. Takže při provedení zbytku algoritmu se musí ještě hodnota paměti přepnout na FALSE. To zaručí, že po zbytek průběhu jednoho impulsu už program nebude danou operaci provádět.

6 Závěr

V této práci jsem vytvořil distribuovaný systém pomocí PLC řady AC500 od firmy ABB. Systém je tvořen jedním nadřazeným PLC, které po komunikační sběrnici komunikuje s dvěma PLC podřazenými a zároveň s ovládací a monitorovací aplikací vytvořenou v programu Control Web 6.

Pro nadřazené PLC jsem zvolil úlohu ovládající model výtahu. Software pro tuto úlohu již byl vytvořen, a tedy bylo třeba k němu dotvořit pouze programový kód pro realizaci komunikace přes protokol Modbus RTU. V tomto kódu se postupně zapínají funkční bloky pro tento protokol, které buď z podřazených PLC čtou, anebo do nich zapisují potřebná data. Tato data mají v paměti PLC přiřazenou adresu tak, aby byla seřazena za sebou, a bylo tak možné je posílat.

Na jednom podřízeném PLC jsem vytvořil algoritmus ovládající úlohy řízení stejnosměrného motoru a tlakovzdušné soustavy. Obě tyto úlohy umožňují ovládání pomocí řídicího napětí i regulaci pomocí zpětné vazby. U stejnosměrného motoru jsem musel vyřešit problém s nepřesným přenosem dat z vysokorychlostního čítače do paměti PLC. Pro odstranění zašumění ze signálu jsem experimentálně navrhnul filtr dolní propust druhého řádu, díky němuž měl signál tvar, se kterým už se dá dále pracovat.

Na druhém podřízeném PLC pak bude realizována úloha pro vrtání plošných spojů navržených v programu Eagle ve verzi 6 a vyšší. Musel jsem zde v prostředí Control Web 6 navrhnout algoritmus, pomocí kterého se do seznamu uloží souřadnice děr, jež mají být vrtány. Vytvořil jsem algoritmus, pomocí kterého se podle nastavených referenčních bodů přepočítají souřadnice děr. Takže plošný spoj může být jakkoliv natočen. Jediné, co se musí dodržet, je položení plošného spoje do portálové frézky tak, aby směrem nahoru byla otočená ta strana plošného spoje, která je zobrazena v ovládací aplikaci.

Komunikaci mezi nadřazeným PLC a programem Control Web jsem vyřešil pomocí přenosu dat přes protokol OPC. Z prostředí CODESYS jsem nejprve musel vybrané proměnné exportovat do souboru. Poté jsem nakonfiguroval OPC server, který je součástí systému CODESYS. V programu Control Web jsem poté musel načtením dat z OPC serveru nakonfigurovat OPC ovladač. Díky tomu jsem mohl vytvořit kanály, pomocí kterých už šlo navázat komunikaci s ovládací a monitorovací aplikací z prostředí Control Web.

Ovládací a monitorovací aplikací obsahuje řídicí prvky pro jednotlivá PLC. Rovněž obsahuje kontrolu připojení PLC. Jedná se o aplikaci v reálném čase. To znamená, že jsem musel v aplikaci dotvořit časování. Toho jsem dosáhl pomocí přístrojů Selector, kde jsem určil podmínky, při jakých se má spustit určitá část skriptu, ovládajícího grafickou vizualizaci aplikace.

Komunikaci v této práci jsem vyřešil pomocí dvou různých protokolů. Systém Control Web 6 je však schopný pracovat i s Modbus protokolem. V současnosti se ale ve škole nenachází licence pro potřebný ovladač. Nicméně vytvořit celý systém na jedné komunikační sběrnici se jeví, spolu s přidáním dalších úloh, jako možnost budoucího řešení.

Seznam použité literatury

- ABB. 1SAP130100R0170 [online]. 2006 [cit. 25.4.2015]. Dostupné z:
<http://www.abb.com/SmartLinks/Default.aspx?gid=ABB1SAP130100R0170>
- ABB. Main catalog [online]. 2014 [cit. 25.4.2015]. Dostupné z:
[http://www09.abb.com/global/scot/scot397.nsf/veritydisplay/f67b3a7c80674dabc1257daa003bd3e5/\\$file/3ADR020077C0202_Industrial%20Automation%20and%20Motion%2012-2014.pdf](http://www09.abb.com/global/scot/scot397.nsf/veritydisplay/f67b3a7c80674dabc1257daa003bd3e5/$file/3ADR020077C0202_Industrial%20Automation%20and%20Motion%2012-2014.pdf)
- BALÁTEŠ, J. *Automatické řízení*. Praha: Nakladatelství BEN, 2003, 654 s. ISBN 80-7300-020-2.
- JANČÍK, L. *Inovace distribuovaného systému řízení v prostředí Control Web 2000*. Ostrava, 2004. 76 s. Bakalářská práce. Strojní fakulta VŠB-TUO. Katedra automatizační techniky a řízení.
- JANEČEK, J. 1993. *Distribuované systémy*. Praha : Vydavatelství ČVUT, 1993.
- KOZIOREK, Jiří a kol. *Distribuované systémy řízení*. Vyd. 1. Ostrava: Vysoká škola báňská - Technická univerzita, 2011, ISBN 978-80-248-2599-1.
- MORAVSKÉ PŘÍSTROJE. *Control Web 6 – nápověda* [software]. [přístup 27.4.2015]. Dostupné z: <http://www.mii.cz/cat?id=68&lang=405>
- NACHTIGAL CH. L. *Instrumentation and Control - Fundamentals and Applications*. New York : John Wiley & Sons, Inc. 1993.
- PAWLENKA, M. *Řízení, monitorování a vzdálená správa laboratorního modelu*: bakalářská práce. Ostrava: VŠB - technická univerzita Ostrava, Fakulta strojní, Katedra automatizační techniky a řízení, 2012, 60 s. Vedoucí práce: Škuta, J.
- RONĚŠOVÁ A. *Přehled protokolu MODBUS* [online]. 2005 [28.4.2015] Dostupné z: <http://home.zcu.cz/~ronesova/bastl/files/modbus.pdf>
- ŠOFER, P. *Využití SCADA/MMI systému pro podporu laboratorních měření*. Ostrava: katedra ATR – 352 VŠB-TUO, 2008, 58 s. Bakalářská práce, vedoucí: Škuta, J.
- ŠOFER, P. *Vizualizační software pro monitorování a konfiguraci systémů s PLC*. Ostrava: katedra ATR – 352 VŠB-TUO, 2011, 66s. Diplomová práce, vedoucí: Tůma, J., konsultant: Škuta, J.

VLACH, J. *Řízení a vizualizace technologických procesů*. PRAHA: BEN, 1999, 160 s. ISBN 80-86056-66-X.

VOJÁČEK, A. Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys). *Automatizace.HW.cz* [online]. HW server s.r.o., 2011 [cit. 25.4.2015]. Dostupné z: <http://automatizace.hw.cz/programovaci-rezimy-pro-plc-dle-iec-611313-codesys>